Web Science

Quiz 1: March 1, 2022

Enter your answers directly into this document (with the exception of #2 and #3).

All answers should be In Your Own Words, using complete sentences with proper spelling and grammar.

Save this document as: answers.docx (or .odf or .pdf) (-5 if wrong name). For all questions other than #2 and #3, you will not receive any credit for answers not placed in this document.

When finished with the quiz, put everything you wrote (this document, all code, etc.) on GitHub into a branch in your lab repo named: quiz1 (-5 if submitted incorrectly). **Do not submit your node\_modules folder! (-15 if you submitted the node\_modules folder)**

1. **Short answers** (25 points): (Answer in complete sentences, explain your answers)
   1. (5) How can I determine the type of device that my page is being displayed on? Give two examples of why I might care.
   2. (5) What is a package-lock.json file? What is it used for? Is it required?

**The package-lock json file contains the exact version of all properties you install from npm in the specified directory. It is used for reproducing the exact same development environment if the code or folder is sent/shipped somewhere. No, it is not required.**

* 1. (5) What is npm? How does it work? Why is it used?

**NPM is the Node Packet Manager. It is used to install applications, modules, and packages. It also manages dependencies. It’s ease of use and primary purpose for NodeJS makes it very popular.**

* 1. (10) Describe **in detail** the sequence(s) of transaction(s) for a frontend to request data from some external entity via Node.

**The frontend makes a request via JS (AJAX & jQuery). Tools such as jQuery can be used to create a get $.get or post $.post request to the node server. The request is usually made to an endpoint set up on Node webserver. When the request reaches the Node server’s endpoint, Node will then take that request and pass it the external API using a variety of methods. Once the method is specified (get or post) to the API, the API will return an object (which is the external API’s response). This object is usually a JSON object to the node server. Once the data is returned the node server, it can be passed to the frontend as the Node server’s API endpoint’s response.**

1. **Coding question**: (40 points) Create a webserver in node.js, name your server: server.js. You may use Express, but you *may not use a generator* – (i.e., NOT express-generator), which will serve a simple frontend (in the technologies of your choosing). The frontend will provide an input field for ZIP code and a series of buttons that issue GET and/or POST requests when clicked to the Node server. (frontend: 10 points)  
     
   Upon entering a ZIP code and clicking the “Temperature” button, your application should send a POST request to <http://localhost:3000/temperature>. Node should then get the current temperature for that ZIP code (I bet you have an API for that!) and send the frontend back that information. The frontend should then output a sentence that says the name of the location and whether it is Freezing (<33F), Cold (between 33 and 50), Warm (between 51 and 80) or Hot (>80) – display the corresponding message in a unique color for each category. (temperature sequence: 10 points)  
     
   Upon clicking the “Is RPI windy?” button, your application should send a GET request to <http://localhost:3000/wind>. Node should get wind speed information for Troy, NY, via that API and send that information back to the frontend. Have the frontend display this information in a unique color. (wind sequence: 10 points)  
     
   Creativity matters; don’t just give me an empty white page with a text entry form box and two buttons. Go beyond the minimum (but remember that creativity doesn’t have to be visual). If you need to, write a short README file that tells me what I should consider for creativity. (creativity: 10 points)  
     
   ***You may use any and all libraries you want for this coding question.***

1. (15) Ensure the package.json file for Q2 has no errors when I run npm install & run your code.
2. (20) Provide **two** different explanations of the code below. The first explanation should be a high-level explanation (no less than four complete sentences) outlining what this code does to someone who has no coding experience. The second explanation should be a *detailed* one explaining line-by-line what the code does. If there are any errors in the code, fix them.

var net = require('net');

var sockets==[];

var s = net.Server(function(socket) {

sockets.push(socket);

socket.on('data', function(d) {

for(var i=0; i<sockets.length;i++) {

if (sockets[i]==socket) continue;

sockets[i].write(d);

}

});

socket.on('end', function() {

var i=sockets.indexOf(socket);

sockets.splice(i,1);

});

});

s.listen(8080);

**High-Level Explanation**

This program establishes a live server connection between the browser and the server. It does this by using the sockets library to establish that connection. Once the connection is established, the program checks the number of sockets running. The program will also write the sockets value into a file. When the socket connection ends, add the value 1 to the array and stop the server connection.

**Detailed Explanation**

-The socket uses the net package from npm. It aliases it as net.

-A variable called sockets is created and holds an array

-A variable called S is created and calls the Server method from the net package

-Within this Server method, an anonymous function is created that adds the current socket to the sockets array

-When the socket from the server is turned on, it will go through the sockets array using a for loop and write the socket value to a file called d.

-When the socket connection ends, an anonymous function is created which houses the variable i. The variable I holds index of the current socket and adds that to index 1 of the sockets array.

-To monitor the webserver, a listening connection is created on port 8080.

1. (+5) What is the name of the RPI-developed chat protocol popular in the 1990s?